Секция 8

БИОИНФОРМАТИКА

А.В. Александров[1], С.В. Казаков[2], С.В. Мельников[3], А.А. Сергушичев[4], Ф.Н. Царев[5], А.А. Шалыто[6]

МЕТОД СБОРКИ КОНТИГОВ ГЕНОМНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ СОВМЕСТНОГО ПРИМЕНЕНИЯ ГРАФОВ ДЕ БРЮИНА И ГРАФОВ ПЕРЕКРЫТИЙ*

Введение

Многие современные задачи биологии и медицины требуют знания геномов живых организмов, которые состоят из нескольких нуклеотидных последовательностей молекул дезоксирибонуклеиновой кислоты (ДНК). Поэтому возникает необходимость в дешевом и быстром методе секвенирования – определения последовательности нуклеотидов в образце ДНК.

Существующие секвенаторы – устройства для чтения ДНК – не позволяют считать за один раз всю молекулу ДНК. Вместо этого они позволяют читать фрагменты генома небольшой длины. В настоящее время получил распространение следующий дешевый и эффективный подход: сначала вычленяется случайно расположенный в геноме фрагмент длиной около 500 нуклеотидов, а затем считываются его префикс и суффикс (длиной порядка 80−120 нуклеотидов каждый). Эти префикс и суффикс называются *парными чтениями*. Описанный процесс повторяется такое число раз, чтобы обеспечить достаточно большое покрытие генома чтениями. Указанным образом работают, например, секвенаторы компании *Illumina* [Illumina, 2012].

[1] *197101, Санкт-Петербург, пр. Кронверкский, д. 49, НИУ ИТМО, alexandrov@rain.ifmo.ru.*
[2] *197101, Санкт-Петербург, пр. Кронверкский, д. 49, НИУ ИТМО, svkazakov@rain.ifmo.ru.*
[3] *197101, Санкт-Петербург, пр. Кронверкский, д. 49, НИУ ИТМО, melnikov@rain.ifmo.ru*
[4] *197101, Санкт-Петербург, пр. Кронверкский, д. 49, НИУ ИТМО, alserg@rain.ifmo.ru*
[5] *197101, Санкт-Петербург, пр. Кронверкский, д. 49, НИУ ИТМО, tsarev@rain.ifmo.ru*
[6] *197101, Санкт-Петербург, пр. Кронверкский, д. 49, НИУ ИТМО, shalyto@mail.ifmo.ru*

* Исследования выполняются в рамках государственных контрактов № 07.514.11.4010 (заключен в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы») и № 16.740.11.0495 (заключен в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы»).

Отметим, что описанные выше префикс и суффикс читаются с разных нитей ДНК: один – с прямой, другой – с обратно-комплементарной, причем неизвестно, который откуда. Поэтому удобно рассматривать геном и чтения, дополненные своими обратно-комплементарными копиями.

Задачей сборки генома является восстановление последовательности ДНК (ее длина составляет от миллионов до миллиардов нуклеотидов у разных живых существ) на основании информации, полученной в результате секвенирования.

1. Предлагаемый метод

Процесс сборки делится, как правило, на следующие этапы:

1. Исправление ошибок в данных секвенирования.
2. Сборка *контигов* − максимальных непрерывных последовательностей нуклеотидов, которые удалось восстановить.
3. Построение *скэффолдов* – последовательностей контигов, разделенных промежутками, для длины которых найдены верхние и нижние оценки.

Одной из наиболее часто используемых при сборке генома математических моделей является, так называемый, граф де Брюина [Pevzner, 1989]. На его использовании основаны следующие программные средства: *Velvet* [Zerbino, 2008], *Allpaths* [Butler, 2008], *AbySS* [Simpson, 2009], *SOAPdenovo* [Li, 2010], *EULER* [Pevzner, 2001].

Одним из недостатков, которым обладают перечисленные программные средства, является большой объем оперативной памяти, необходимый им для сборки генома, сходного по размерам с геномом человека (2–3 миллиарда нуклеотидов). Так, например, *SOAPdenovo* необходимо порядка 140 ГБ оперативной памяти, а *ABySS* – 21 компьютер с 16 ГБ каждый (всего – 336 ГБ). Такие затраты памяти обусловлены наличием ошибок секвенирования в исходных данных (такие ошибки ведут к увеличению размера графа де Брюина), а также неэкономным методом хранения этого графа.

В настоящей работе предлагается метод, лишенный указанного недостатка. Построение скэффолдов в настоящей работе не рассматривается.

Сборка контигов в предлагаемом методе выполняется в два этапа:

1. **Сборка квазиконтигов** из чтений геномной последовательности. Квазиконтигами называются последовательности, которые, с одной стороны, длиннее чтений, но, с другой стороны, не являются контигами в смысле невозможности наращивания вправо и влево. Этот этап выполняется с использованием графа де Брюина.
2. **Сборка контигов из квазиконтигов.** Выполняется с использованием графа перекрытий и метода *Overlap-Layout-Consensus*.

2. Сборка квазиконтигов из чтений геномной последовательности

В предлагаемом методе используется граф де Брюина, в котором множество ребер состоит только из «надежных» (k+1)-меров – тех, которые встречаются в чтениях достаточно большое число раз, не меньшее некоторого порогового значения, для того чтобы их можно было с очень большой вероятностью считать входящими в геном. Множество вершин состоит из тех вершин графа де Брюина, которым инцидентно хотя бы одно из выбранных ребер. Если участок нуклеотидной последовательности покрылся достаточно хорошо, то все входящие в него (k+1)-меры по много раз входят в исходные данные, а тогда в этом графе существует путь между первым и последним k-мерами участка.

Предлагаемый метод основан на поиске такого пути для фрагмента, соответствующего парным чтениям. Из всех путей нас интересуют только те, которые укладываются в априорные границы длин фрагментов, поэтому слишком короткие и слишком длинные пути можно отбросить. Если нашелся единственный путь, то можно с очень большой уверенностью сказать, что он соответствует реальной подстроке геномной последовательности, поэтому этот фрагмент считается восстановленным, а найденный путь выводится.

Для того чтобы потребление памяти при применении предлагаемого метода было не очень большим, необходимо иметь компактное представление используемого подграфа графа де Брюина. Для этого достаточно хранить только множество его ребер, что можно эффективно делать, используя, например, хеш-таблицу с открытой адресацией. Преимуществами такого подхода хранения перед другими являются его простота, быстродействие и возможность балансировки между используемой памятью и скоростью. Более эффективными, с точки зрения потребляемой памяти, являются rank/select словари [Okanohara, 2006], которые позволяют сделать ее использование близким к энтропии, но из-за этого увеличивается время доступа.

Важным является и то, что каждый (k+1)-мер входит в граф вместе с обратно-комплементарным. Тогда вместо пары (k+1)-меров s и s^{rc} можно хранить только один, определяемый по некоторому правилу – например, таким правилом может быть выбор лексикографически минимального (k+1)-мера. В этом случае необходимый объем памяти уменьшается примерно в два раза для четных k и ровно в два раза – для нечетных (только для четных k бывают обратно комплементарные себе (k+1)-меры).

3. Сборка контигов из квазиконтигов

Сборка контигов из квазиконтигов основана на подходе Overlap-Layout-Consensus [Bockenhauer, 2007] и состоит из нескольких этапов:

1. Построение графа перекрытий между квазиконтигами.

2. Уточнение графа перекрытий.
3. Поиск контигов в графе перекрытий.

Для поиска перекрытий построим строку вида C1$C2$C3$...Cn$, где Ci – i-ый квазиконтиг, а n – число квазиконтигов. После этого необходимо построить суффиксный массив [Гасфилд, 2003] для этой строки – отсортированный массив всех суффиксов строки. С его помощью можно найти все квазиконтиги, в которых встречается заданная подстрока. Это можно сделать, например, с помощью бинарного поиска суффиксов в суффиксном массиве, которые начинаются с заданной подстроки. Они будут располагаться рядом за счет сортировки.

Зафиксируем квазиконтиг, все перекрытия с которым требуется найти. Будем рассматривать его префиксы в порядке увеличения длины, начиная с минимального порога. Для каждого префикса будем проверять − входит ли такая подстрока с добавленным в конце $ в суффиксный массив. Для того чтобы учесть еще и неточные перекрытия, проверяются не только сами префиксы, но и префиксы, в которые внесены небольшие изменения.

На следующем этапе происходят анализ найденных перекрытий, а также их добавление и удаление. Добавление происходит в случае, если квазиконтиг A перекрывается с квазиконтигами B и C, причем, квазиконтиги B и C должны перекрываться, но такое перекрытие найдено не было. Удаление происходит, если квазиконтиг A перекрывается с квазиконтигом B, но B не похож на большинство квазиконтигов, с которыми перекрывается A.

Для поиска контигов выполняется поиск в ширину [Кормен, 2011] в графе перекрытий. Он прерывается, если после текущего квазиконтига нет консенсуса, что означает, что квазиконтиги, перекрывающиеся с ним, различаются в большом числе позиций.

4. Экспериментальные исследования

Экспериментальные исследования разработанного метода проводились в рамках проекта Assemblathon 2 [Assemblathon, 2012], организованного Калифорнийским университетом в Дэвисе (University of California, Davis), на одном из наборов данных, который был подготовлен организаторами, – наборе чтений рыбы Maylandia zebra. Размер генома этой рыбы оценивается примерно в один миллиард нуклеотидов.

Для сборки контигов использовался набор чтений со средним размером фрагмента 180 и 60-кратным покрытием. Общий объем исходных данных составлял 140 ГБ (в сжатом виде), из них авторами были использованы только 44 ГБ.

Алгоритмы сборки генома были реализованы на языке программирования Java. Для запуска программ использовался компьютер с 32 ГБ оперативной памяти и двумя 4-ядерными процессорами. Суммарное время

работы всех трех этапов – исправления ошибок, сборки квазиконтигов и сборки контигов – составило пять суток. Опишем подробнее результаты каждого из этапов.

Перед исправлением ошибок чтения были обрезаны, чтобы вероятность отдельной ошибки в каждом нуклеотиде не превышала 10 %. После этого длина всех чтений в среднем уменьшилась на 20 %. Исправление ошибок работало в течение 42 часов. В результате было найдено 150 миллионов исправлений. Всего чтений было 600 миллионов, поэтому было исправлено в среднем каждое четвертое чтение. Сборка квазиконтигов заняла 38 часов. Квазиконтиги были получены из 60 % чтений. Сборка контигов выполнилась за 26 часов. В результате было получено 734165 контигов, суммарный размер которых составляет 680106 нуклеотидов. Длина максимального составляет 23514 нуклеотидов, средняя длина – 927, значение метрики N50 – 1799.

Заключение

Предложен метод сборки контигов геномных последовательностей, основанный на совместном использовании графа де Брюина и графа перекрытий. Экспериментальное исследование этого метода проведено в рамках проекта Assemblathon 2. Это экспериментальное исследование показало, что с помощью разработанного метода можно собирать геномы размером в миллиард нуклеотидов быстрее чем за неделю.

Библиографический список

1. **[Assemblathon, 2012]** The Assemblathon – http://www.assemblathon.org.
2. **[Bockenhauer, 2007]** Bockenhauer H.-J. and Bongrratz D. Algorithmic Aspects of Bioinformatics. – Springer, 2007.
3. **[Butler, 2008]** Butler J., MacCallum I., Kleber M., Shlyakhter I.A., Belmonte M.K., Lander E.S., Nusbaum C., Jaffe D.B. ALLPATHS: De novo assembly of wholegenome shotgun microreads // Genome Research. – 2008. – Vol. 18. – P. 810-820.
4. **[IHGSC, 2001]** International Human Genome Sequencing Consortium. 2001. Initial sequencing and analysis of the human genome // Nature. – Vol. 409, № 6822. – P. 860-921.
5. **[Illumina, 2012]** Illumina, Inc. – http://www.illumina.com/.
6. **[Li, 2010]** Li R., Zhu H., Ruan J., Qian W., Fang X., Shi Z., Li Y., Li S., Shan G., Kristiansen K., et al. De novo assembly of human genomes with massively parallel short read sequencing // Genome Research. – 2010. – Vol. 20. – P. 265-272.
7. **[Okanohara, 2006]** Okanohara D., Sadakane K. Practical entropy-compressed rank/select dictionary // Computing Research Repository. 2006. http://arxiv.org/abs /cs/0610001.
8. **[Pevzner, 1989]** Pevzner P.A. 1-Tuple DNA sequencing: computer analysis // J. Biomol. Struct. Dyn. – 1989. – Vol. 7. – P. 63-73.
9. **[Pevzner, 2001]** Pevzner P.A., Tang H., Waterman M. S. EULER: An Eulerian path approach to DNA fragment assembly // Proc. Natl. Acad. Sci. – 2001. – № 98. – P. 9748-9753.

10. **[Simpson, 2009]** Simpson J.T., Wong K., Jackman S.D., Schein J.E., Jones S.J., Birol I. ABySS: A parallel assembler for short read sequence data // Genome Research. – 2009. – Vol. 19. – P. 1117-1123.
11. **[Zerbino, 2008]** Zerbino D.R., Birney E. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. // Genome Research. – 2008. – Vol. 18. – P. 821-829.
12. [**Гасфилд, 2003**] Гасфилд Д. Строки, деревья и последовательности в алгоритмах. Информатика и вычислительная биология. – СПб.: Невский диалект, 2003.
13. [**Кормен, 2011**] Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. – М.: Вильямс, 2011.